Question 1
Please give short answers (1-3 sentences) to the following questions.

(a) What is the *curse of dimensionality*?

(b) What is *overfitting* and how can we avoid it?

(c) It is often assumed that the magnitude (i.e., $|w_1|$, $|w_2|$, ..., , $|w_D|$) of the linear regression coefficients $\mathbf{w} = [w_1, \ldots, w_D]$, where $y = \mathbf{w}^T\mathbf{x}$, indicates the importance of the corresponding features. Describe a situation where this may not be true.

Question 2
Please select the correct answer(s) to the following questions. Multiple answers could be correct.

(a) Which are possible hyper-parameters?

A. The learning rate α of gradient descent

B. The weight of regularization λ in linear regression

C. The degree of polynomial in non-linear regression with a polynomial function

D. The weights w in non-linear regression

Question 3

Assume a non-linear regression, whose output $y \in \mathbb{R}$ is predicted from the input $x \in \mathbb{R}$ as follows:

$$y = f(x) = bx^3 + c$$

The goal of the non-linear regression is to learn the weights $b, c \in \mathbb{R}$ from the training data $\mathcal{D} = \{(x_1, y_1), \ldots, (x_N, y_N)\}$.

Hint: For the following, you can use the formula: $(z_1 + z_2 + z_3)^2 = z_1^2 + z_2^2 + z_3^2 + 2z_1z_2 + 2z_2z_3 + 2z_1z_3$

(a) Write the residual sum of squares (RSS) error between the actual and predicted outcomes.

(b) Assuming that c is constant, derive the gradient descent formula for updating the weight $b \in \mathbb{R}$ so that the RRS error (as obtained from question a) is minimized with respect to b. Give the batch update formula and the stochastic update for a random sample (x_m, y_m). Set up your formulas so that the learning rate parameter is $\nu(k) > 0$, in which k is the iteration index.

(c) Assuming that b is constant, derive the gradient descent formula for updating the weight $c \in \mathbb{R}$ so that the RRS error (as obtained from question a) is minimized with respect to c. Give the batch update formula and the mini-batch update for a random subsample of the training data $\mathcal{D}_s \in \mathcal{D}$. Set up your formulas so that the learning rate parameter is $\nu(k) > 0$, in which k is the iteration index.

(d) We now apply $l2$-norm regularization for the aforementioned non-linear regression in terms of both weights $b, c \in \mathbb{R}$. Write the expression of the evaluation criterion for the non-linear regression with regularization, including the RSS error and regularization term, assuming $\lambda > 0$ as the model complexity for both b and c.

(e) Derive the gradient descent formula for jointly updating weights $b, c \in \mathbb{R}$ so that the optimization criterion of the regularized non-linear regression (as obtained from question d) is minimized jointly with respect to b and c. Give the batch update formula using the learning rate parameter $\nu(k) > 0$, in which k is the iteration index.
Hint: Assume a weight vector $\mathbf{w} = [b, c]^T \in \mathbb{R}^2$.

Question 4

Let the random variable $\mathcal{X}$ follow an exponential distribution with parameter λ, i.e., the probability of x is $f(x) = \lambda e^{-\lambda x}$. Also let's assume a set of independent and identically distributed data (i.i.d.) samples $\mathcal{X} = \{x_1, \ldots, x_N\}$ which follows the exponential distribution

(a) Find the log-likelihood of the data.

(b) Find the extreme point of the above likelihood with respect to λ.

Question 5

Assume that you have a set of training and test data, $\mathbf{X}^{\mathbf{train}} \in \mathbb{R}^{D \times N_1}$ and $\mathbf{X}^{\mathbf{test}} \in \mathbb{R}^{D \times N_2}$, respectively, with corresponding label vectors $\mathbf{y}^{\mathbf{train}} \in \mathbb{R}^{N_1}$ and $\mathbf{y}^{\mathbf{test}} \in \mathbb{R}^{N_2}$. You would like to perform classification using logistic regression and use cross-validation on the training set to determine the best value of the learning rate $\alpha = \{0.001, 0.01, 0.1\}$ for the gradient descent optimization of logistic regression. Please provide a basic pseudocode to do this. In the pseudocode, please also include the last evaluation step after having identified the best learning rate α based on the test data.

Hint: You can assume a function *pred*=LR(X,y,Z,α), which provides a decision for test samples Z using training data X, training labels y and gradient descent learning rate α, and a function *acc*=ComputeAcc(*pred*,*lab*), which computes the classification accuracy between predicted class *pred* and actual labels *lab*.